**Engineer**

# Brannmur

regelsettet som avgjør hvilken trafikk som slipper gjennom

## Hva det er

Mellom en tjeneste og nettverket den lytter på, sitter ofte en brannmur: et filter på nettverksveien som sammenligner hver pakke eller forbindelse med en liste regler og deretter lar den passere, forkaster den eller svarer med en avvisning. Den kan være programvare på maskinen den beskytter, en egen boks mellom to nettverk eller en tjeneste som en skyleverandør kjører foran en virtuell maskin. Mange sier firewall også på norsk, Linux-administratorer sier gjerne pakkefilter om det samme, og skyleverandører selger brannmurer under navn som sikkerhetsgruppe (security group). Brannmurer kommer som regel i lag: én hos leverandøren og én på verten.

NISTs veiledning på området, SP 800-41 Rev. 1 fra september 2009, sorterer dem etter hvor mye av trafikken de leser. Et pakkefilter ser på adresser, protokoll og portnumre, én pakke om gangen. En tilstandsbasert brannmur (stateful firewall) husker i tillegg forbindelser, så svaret på en forespørsel serveren selv sendte, slipper inn uten en egen regel. En applikasjonsproxy (application-proxy gateway) avslutter forbindelsen og leser protokollen inne i den. En vertsbasert brannmur kjører på serveren eller den bærbare maskinen den vokter, og vet hvilket program som eier en socket.

På en Linux-server skjer filtreringen i netfilter, rammeverket for pakkefiltrering i kjernen. Siden kjerneversjon 3.13 har nftables vært tilgjengelig som erstatning for de eldre iptables-verktøyene, og grensesnitt som ufw skriver reglene for folk som heller vil si hva de ønsker, enn å skrive kjeder. Uansett hva som skriver reglene, er modellen den samme: en ordnet liste som leses ovenfra, der den første regelen som treffer, avgjør.

En brannmur blir ofte blandet sammen med verktøy i nærheten. En brannmur for webapplikasjoner (WAF) leser HTTP-forespørsler og blokkerer angrepsmønstre i dem; en nettverksbrannmur ser aldri så dypt. Et verktøy som utestenger inntrengere, leser logger og legger til midlertidige regler mot adresser som oppfører seg dårlig, og er dermed avhengig av en brannmur under seg. Et VPN avgjør hvem som i det hele tatt er på nettverket. Hvert av dem svarer på et annet spørsmål, og den enkle regellisten blir ikke overflødig av noen av dem.

Bruk begrepet i en beslutning

Bruk «Hva er en brannmur?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Kontoer, passord og tilganger ordnet slik at folk har nøyaktig den tilgangen jobben krever, og de som slutter, mister den samme dag.

**Se tjenestene våre** Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Skrev 1 000 RBAC-regler for å sikre Linux-servere — 1,000 tilgangsregler skrevet for å tette et sikkerhetshull

**Skrev 1 000 RBAC-regler for å sikre Linux-servere** Ta kontakt →

## Hvorfor det er viktig

Hver tjeneste som lytter på en offentlig adresse, er en dør, og brannmuren er fortegnelsen over hvilke dører som finnes. Avgjørelsen et team faktisk står overfor, har tre deler: hvor brannmuren skal ligge, hva den tillater som standard, og hvem som har lov til å endre den.

En brannmur i skyen hos leverandøren stopper trafikken før den når maskinen, fortsetter å virke selv om noen får root på verten, og koster ikke serveren noe CPU-tid. Reglene ligger i en konsoll eller bak et API, utenfor serverens egen konfigurasjon, så de glir ut av takt med alt annet med mindre de er skrevet som kode. En brannmur på verten følger med maskinen, kan sette en hastighetsbegrensning på én enkelt nettverksport og kan testes fra maskinen selv, men en kompromittert root-konto kan skrive den om, og det kan også programvare som styrer sine egne regler. Å kjøre begge er vanlig og fornuftig, til prisen av to lister som må holdes i takt: hver port som åpnes eller lukkes, er en endring to steder.

Standardvalget er den andre delen. NIST anbefaler å avvise all trafikk som ikke uttrykkelig er tillatt, og for innkommende trafikk er nesten alle enige. Utgående filtrering er der avveiningen merkes: den begrenser hva en kompromittert tjeneste kan nå, og den stopper pakkeoppdateringer, fornyelse av sertifikater og overvåking helt til hver destinasjon har fått sin regel.

Den tredje delen er eierskap. En brannmur som avviser alt som standard, er bare så riktig som listen over porter noen har deklarert, så det nyttige dokumentet er nettopp den deklarerte listen, holdt under versjonskontroll og sjekket mot reglene som faktisk er aktive. Programvaren er gratis på hver vanlig Linux-distribusjon; det som koster, er oppmerksomheten det krever å holde deklarasjonen sann. Og en brannmur er en andre forsvarslinje. En tjeneste som er bundet til loopback-adressen, kan ikke nås utenfra uansett hva reglene sier, og det er en sterkere garanti enn en avvisningsregel foran en tjeneste som lytter på alle adresser.

## Slik ser det ut i praksis

Tre deler av selskapets infrastrukturarbeid viser brannmuren fra tre sider: hvordan den sjekkes, når den opprettes, og hva som kan gå rundt den.

På én vert sto en hastighetsbegrensning for SSH-porten i regellisten i et år uten at en eneste pakke nådde den, fordi en bredere regel som tillot trafikken, sto over den og slo til først. Statusverktøyene viste begrensningen hele året, og de hadde rett i det: den fantes. Sjekkene ble skrevet om for å lese plassering i stedet for tilstedeværelse: de sammenligner nå rekkefølgen i den nummererte regellisten, og i kjeden kjernen faktisk filtrerer med, mot det som var tenkt. Hver port får én regel, og hver regel navngir en protokoll; SSH har en hastighetsbegrensning og ingen egen allow-regel. De fem offentlige portene er deklarert samlet og sjekkes som en helhet, så en åpen port som deklarasjonen utelater, får sjekken til å feile, og sikkerhetsrapporten viser reglene i den rekkefølgen nettverkspakkene møter dem, med en merknad om hvordan rekkefølgen skal leses.

Den første serveren ble bygd for hånd og fikk brannmuren sin først senere, og det etterlot et vindu der en fersk maskin kunne nås fra internett. For en server nummer to hos en annen leverandør opprettes brannmuren av sitt eget play før maskinen, så den nye maskinen møter internett med reglene allerede på plass. Den leverandøren knytter brannmurer til maskiner gjennom tagger, så taggene opprettes først: en regel som retter seg mot en tagg, trenger at taggen finnes. Brannmuren hos hver leverandør er skrevet direkte mot leverandørens REST-API med en vanlig HTTP-oppgave. Oppgaven gjør ingenting i en tørrkjøring, så endringer øves på selve brannmur-playet, siden en brannmur hos leverandøren ikke koster noe å opprette og kan slettes igjen.

Den tredje gjelder containere. Docker skriver sine egne oppføringer i kjernens pakkefilter, og en port Docker publiserer for en container, blir matchet før noe av det herdingsrollen på verten har lagt inn, så en container kan nås utenfra uansett hva brannmuren har fått beskjed om. Containerne på denne plattformen kjører under systemd gjennom Podman og Quadlet, deler vertens nettverk og publiserer ingen porter, så det finnes ingen containerport som noe kan bruke til å gå rundt reglene.

## Hvor det går galt

**En regel står i statusutskriften og slår aldri til.** Reglene evalueres i rekkefølge, og første treff vinner, så en bred allow-regel plassert over en smalere begrensning eller avvisning skygger helt for den. Listen viser fortsatt begge. Les reglene slik kjernen gjør, nummerert og i rekkefølge, sett spesifikke regler over generelle, og test fra en maskin utenfor:

```
ufw status numbered
nft list ruleset
```

**En container svarer på en port brannmuren avviser.** Docker publiserer porter ved å omdirigere trafikk i kjernens NAT-tabell, før den når kjedene ufw filtrerer, og Dockers egen dokumentasjon sier at dette i praksis går utenom brannmurkonfigurasjonen. Publiser bare til loopback-adressen, sett en brannmur hos leverandøren foran, eller kjør containere uten publiserte porter, slik det er beskrevet ovenfor.

**IPv4 er låst ned, og IPv6 står vidt åpen.** De to adressefamiliene har hvert sitt regelsett i iptables, og ufw skriver bare IPv6-regler når IPv6 er slått på i filen med standardinnstillinger. En vert med en offentlig IPv6-adresse og et regelsett bare for IPv4 er eksponert på hver port den lytter på. Bruk en nftables-tabell i familien `inet`, som ser begge, eller gå gjennom begge listene, og skann vertens IPv6-adresse utenfra.

**SSH-økten fryser idet brannmuren slås på.** Standardavvisningen ble aktiv før regelen for SSH, eller daemonen for SSH ble flyttet til en annen port mens regelen fortsatt nevner 22. Legg inn SSH-regelen først, hold leverandørens konsoll åpen i et annet vindu, og planlegg en automatisk tilbakerulling noen minutter fram i tid, som avbrytes så snart en ny økt kommer inn.

**En regel åpner mer enn den som skrev den, mente.** I ufw åpner en regel som navngir en port uten protokoll, porten for både TCP og UDP. Navngi protokollen i hver regel, og deklarer den forventede flaten som en helhet, slik at alt ekstra får en sjekk til å feile i stedet for å vente på en revisjon.

## Les mer

- Firewall — definisjonene NIST-publikasjonene bruker, hver lenket til kildedokumentet sitt.
- SP 800-41 Rev. 1: Guidelines on Firewalls and Firewall Policy — typer brannmurer, en policy som avviser alt som ikke er tillatt, og forvaltning av regler, fra NIST.
- ufw(8) — brannmurgrensesnittet i Ubuntu, bl.a. rekkefølgen på reglene og regelen `limit`.
- nftables wiki — netfilter-prosjektets dokumentasjon av rammeverket som erstatter iptables.
- Packet filtering and firewalls — Dockers egen forklaring på hvordan publiserte porter går utenom vertens regler.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

**Se hva det koster** Ta kontakt →

# Gjør begrepene praktiske

Spør hvordan «Hva er en brannmur?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/firewall/